Program No.: 05

**Aim:**

Use Strings and develop a python application and analyze various string patterns.

**Topics covered:**

Handling Strings using Python.

**Course Outcome**

CO1: Implement the basic concepts of Python.

Cognitive Level: K3

**Case Studies:**

***Case Study 01:***

***Problem Statement:***

Python script to reverse a string, check whether it is a palindrome or not and delete a string.

***Problem Solution:***

1. Take a string as input.

2. Use slice operator [::-1] to reverse the string and store it in another variable.

3. If 2 strings are equal, print it is a palindrome.

4. Else print it is not a palindrome.

5. Delete the given string using del statement

6. Exit.

***Program/Source Code:***

#Python file to reverse a string and check if it is palindrome.

"""

Case study : 01

File name : cse1.py

Topics : Strings

"""

#Taking input

str=input(“Enter a string:”)

#logic

str1=str[::-1]

if(str==str1):

print(“Given string is a palindrome”)

else:

print(“Given string is not a palindrome”)

del str

print(“Given string is deleted using del statement”)

***Program Explanation:***

1. Take a string as input and store in str.

2. Use slice operator [::-1] to reverse the string and store it in another variable str1.

3. If str is equal to str1, print it is a palindrome

4. Else print it is not a palindrome.

5. Delete the given sting using del statement.

6.Exit.

***Runtime Test Cases:***

1. Enter a String: script

Given string is not a palindrome

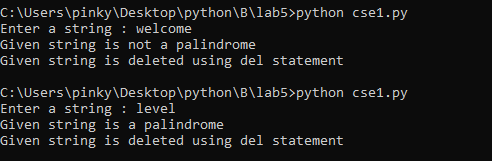
Given string is deleted using del statement

1. Enter a String: tit

Given string is a palindrome

Given string is deleted using del statement

**Output:**



**Result:**

Python script to find the string is palindrome or not is done successfully.

***Case Study 02:***

***Problem Statement:***

Given two strings of lowercase English letters, A and B, perform the following operations: Sum the lengths of A and B.

1. Determine if A is lexicographically larger than B (i.e.: does B come before A in the dictionary?).

2. Capitalize the first letter in A and B prints them on a single line, separated by a space.

***Problem Solution:***

1. Take two strings as input

2. Use len() function to find the length of 2 strings ,add and print it.

3. For values in two strings do,

4. If ascii value of 1st string is equal to 2nd string.

5. If ascii value of 1st string is greater than 2nd, print str1 is greater and break the loop with p=1

6. Else break the loop with p=0

7. if p=0 print str1 is not greater

8. Exit.

***Program/Source Code:***

#Practice program to check for lexicography and capitalize the strings.

"""

Case study : 02

File name : cse2.py

Topics : Strings

"""

# take input from the user

str1=input("Enter 1st string:")

str2=input("Enter 2nd string:")

#logic

p=0

print(“sum of length of 2 strings is”,len(str1)+len(str2))

for i in range(0,len(str1)):

if(ord(str1[i])==ord(str2[i])):

Continue

if(ord(str1[i])>ord(str2[i])):

p=1

break

else:

break

if(p==1):

print(“str1 is greater than str2”)

else:

print(“str1 is not greater than str2”)

print(str1.capitalize(),“”,str2.capitalize())

***Program Explanation:***

1. Take two strings as input

2. Use len() function to find the length of 2 strings, add and print it.

3. For values in 2 strings do.

4. if ascii value of 1st string is equal to 2nd continue

5. if ascii value of 1st string is greater than 2nd print str1 is greater and break the loop with p=1

6. else break the loop with p=0

7. if p=0 print str1 is not greater

8. Exit.

***Runtime Test Cases:***

1. Enter 1st string : hi

Enter 2nd string : kundana

Sum of lengths of 2nd string is 9

Str1 is not greater than str2

Hi Kundana

1. Enter 1st string : gud

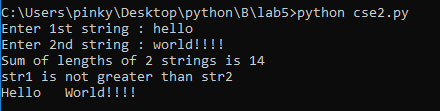
Enter 2nd string : morning

Sum of lengths of 2nd string is 10

Str1 is not greater than str2

Gud Morning

**Output:**



**Result:**

Python program to check for lexicography and capitalize the strings is done successfully.

***Case Study 03:***

***Problem Statement:***

Given a string, a, and two indices, start and end, print a substring consisting of all characters in the inclusive range from start to end-1.

***Problem Solution:***

1. Take a string as inputs.

2. taking stating and ending indices as input.

3. print string with slice operator[starting index : ending index] .

4. Exit.

***Program/Source Code:***

#Practice scripts to use slice operator

"""

Case study : 03

File name : cse3.py

Topics : Strings

"""

# taking input

str=input(“Enter a string:”)

m=int(input(“Enter starting index:”))

n=int(input(“Enter ending index:”))

#printing

print(“Required string is:”,str[m:n])

***Program Explanation:***

1. Take a sting as input into str.

2. Taking starting and ending indices as input into m and n

3. Print string with slice operator[m:n]

4. Exit.

***Runtime Test Cases:***

1. Enter a string: Helloworld

Enter starting index: 3

Enter ending index: 7

Required string is: lowo

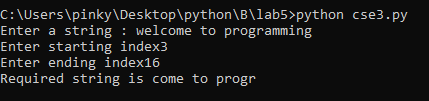
1. Enter a string: Happy Birthday

Enter starting index: 2

Enter ending index: 8

Required string is: ppy Bi

**Output:**



**Result:**

Python script to use slice operator is successfully done.

***Case Study 04:***

***Problem Statement:***

Two strings, a and b, are called anagrams if they contain all the same characters in the same frequencies. For example, the anagrams of CAT are CAT, ACT, TAC, TCA, ATC, and CTA.

If a and b are case-insensitive anagrams, print "Anagrams"; otherwise, print "Not Anagrams" instead.

***Problem Solution:***

1. Take two strings as inputs.

2. Use lower() function to change strings to lowercase.

3. Declare two temporary arrays to store the count of alphabets.

4. If both lengths are different print NOT ANAGRAMS.

5. Else, for both strings check every character and store the count of alphabet in temporary arrays.

6. if count of every character is equal in both the arrays print ANAGRAMS.

7. Else print NOT ANAGRAMS.

8. Exit.

***Program/Source Code:***

#Practice scripts to check whether the given number is even or odd

"""

Case study : 04

File name : cse4.py

Topics : Strings

"""

# taking input

str1=int(input(“Enter 1st sting:”))

str2=int (input(“Enter 2nd sting:”))

str3=str1.lower()

str4=str2.lower()

#logic

arr1=[0]\*26

arr2=[0]\*26

if(len(str3)!=len(str4)):

print (“Not Anagrams”)

else:

p=0

for i in str3:

arr1[ord(i)-97]+=1

for i in str4:

arr2[ord(i)-97]+=1

for i in range(0,len(str3)):

if(arr1[i]!=arr2[i]):

p=1

break

if(p==1):

print(“Not Anagrams”)

else:

print(“Anagrams”)

***Program Explanation:***

1. Take two strings as inputs.

2. Use lower() function to change strings to lowercase.

3. Declare two temporary arrays arr1 and arr2 to store the count of alphabets.

4. If both lengths are different print NOT ANAGRAMS.

5. Else, for both strings check every character and store the count of alphabet in temporary arrays.

6. if count of every character is equal in both the arrays print ANAGRAMS.

7. Else print NOT ANAGRAMS.

8. Exit.

***Runtime Test Cases:***

1. Enter 1st string: Anagram

Enter 2nd string: Margana

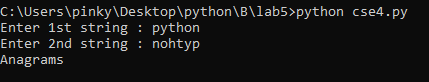
Anagrams

1. Enter 1st string: hello

Enter 2nd string: hillo

Not Anagrams

**Output:**



**Result:**

The python script to check given strings are anagrams or not is successfully done.

***Case Study 05:***

***Problem Statement:***

A password is said to be strong if it satisfies the following criteria:

1. It contains at least one lowercase English character.

2. It contains at least one uppercase English character.

3. Its length is at least 8.

4. It contains at least one digit.

5. Given a string, find its strength. Let a strong password is one that satisfies all above conditions. A moderate password is one that satisfies first three conditions and has length at least.

6. Otherwise password is weak.

***Problem Solution:***

1. Take a string as inputs.

2. for value in string, do

3. if ascii value of character is between 96 and 123 take lo=1.

4. if ascii value of character is between 64 and 91 take up=1.

5. if ascii value of character is between 47 and 58 take num=1

6. if lo, up, num are 1 and length of string is >=8 print strong

7. else if lo, up, num are 1 and length of string is >=6 print medium

8. else print weak

9. Exit.

***Program/Source Code:***

#Practice scripts to check whether password is strong or not.

"""

Case study : 05

File name : cse5.py

Topics : Strings

"""

#taking input

str=input(“Enter password:”)

#logic

up=0

lo=0

num=0

for i in str:

if(ord(i)>96 and ord(i)<123):

lo=1

if(ord(i)>64 and ord(i)<91):

up=1

if(ord(i)>47 and ord(i)<58):

num=1

if(lo==1 and up==1 and num==1 and len(str)>=8):

print(“Password is strong”)

elif(lo==1 and up==1 and num==1 and len(str)>=6):

print(“Password is medium”)

else:

print(“Password is weak”)

***Program Explanation:***

1. Take a string as input.

2. for values in string, do

3. if ascii value of character is between 96 and 123 take lo=1

4. if ascii value of character is between 64 and 91 take up=1

5. if ascii value of character is between 47 and 58 take num=1

6. if lo, up, num are 1 and length of string is >=8 print strong

7. else if lo, up, num are 1 and length of string is >=6 print medium

8. else print weak

9. Exit

***Runtime Test Cases:***

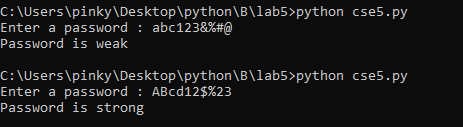
1. Enter Password: abc123#$

Password is weak

1. Enter Password: Abc123#$

Password is strong

**Output:**



**Result:**

The python script to check whether the password is strong or not is successfully done.

***Case Study 06:***

***Problem Statement:***

A string is said to be complete if it contains all the characters from a to z. Given a string, check if it complete or not.

***Problem Solution:***

1. Take a number of test cases as input.

2. Take that number of strings as input and store them in a string array

3. For every character in each string take the ascii value and decrease it by 97 and store it in a temporary array which stores the count of alphabets

4. If all values in temporary array are >=1 store YES in out array

5. else store NO

6. print out elements

7. Exit.

***Program/Source Code:***

#Practice scripts to check whether string has all letters or not.

"""

Case study : 06

File name : cse6.py

Topics : Strings

"""

#taking input

n=int(input(“Enter number of test cases :”))

str=[None]\*n

out=[None]\*n

for i in range(0,n):

str[i]=input(“Enter a string:”)

arr=[0]\*26

for j in str[i]:

arr[ord(j)-97]+=1

p=0

for j in range(0,26):

if(arr[j]==0):

p=1

break

if(p==0):

out[i]=”YES”

else:

out[i]=”NO”

for i in range(0,n):

print(out[i])

***Program Explanation:***

1. Take a number of test cases as input.

2. Take that number of strings as input and store them in a string array.

3. For every character in each string take the ascii value and decrease it by 97 and store it in a temporary array which stores the count of alphabets.

4. If all values in temporary array are >=1 store YES in out array.

5. Else store NO

6. print out elements.

7. Exit.

***Runtime Test Cases:***

1. Enter Number of test cases :3

Enter a string: wyyga

Enter a string: qwertyuioplkjhgfdsazxcvbnm

Enter a string: qwerty

NO

YES

NO

2. Enter Number of test cases :2

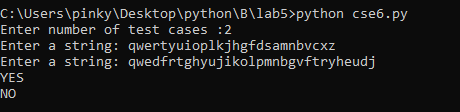
Enter a string: abcdef

Enter a string: ifjklmnopqrs

NO

NO

**Output:**



**Result:**

The python script to check whether string has all letters or not is successfully completed.

***Case Study 07:***

***Problem Statement:***

You are given a string S, which comprises English alphabets, determine the count of all the vowels in S. Vowels are [a, i, e, o, u]. Print the count of all the vowels that are available in the string S.

***Problem Solution:***

1. Take a string as input.

2. turn it into lowercase using lower() function.

3. For every character in string if it is a or e or I or o or u increase the count of respective var.

4. print all the counts.

5. Exit.

***Program/Source Code:***

#Python script to check how many vowels are present.

"""

Case study : 07

File name : cse7.py

Topics : Strings

"""

#Taking input

str=input(“Enter a string:”)

str.lower()

a=0

e=0

i=0

o=0

u=0

for j in str:

if(j==‘a’):

a+=1

if(j==‘e’):

e+=1

if(j==‘i’):

i+=1

if(j==‘o’):

o+=1

if(j==‘u’):

u+=1

print(“a:”,a)

print(“e:”,e)

print(“i:”,i)

print(“o:”,o)

print(“u:”,u)

***Program Explanation:***

1.Take a string as input.

2. turn it into lowercase using lower() function.

3. for every character in string, if it is a or e or i or o or u increase the count of respective var

4. print all the count.

5. Exit.

***Runtime Test Cases:***

1. Enter a string: gud morning. Nice to see u

a: 0

e: 3

i: 2

o: 2

u: 2

1. Enter a string::aeiou aeiou hahaha

a: 5

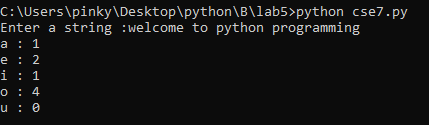
e: 2

i: 2

o: 2

u: 2

**Output:**



**Result:**

The python script to check how many vowels are present. is successfully done.

***Case Study 08:***

***Problem Statement:***

You are given a string S. Count the numbers of occurrences of all the digits in the string S.

***Problem Solution:***

1. Take a string as input.

2. take temp array to store count of numbers.

3. For every character in string, if ascii value is between 47 and 58 increase count of acsii value –48 in array.

4. print all array elements.

5. Exit.

***Program/Source Code:***

#Python program to Count the numbers of occurrences of all the digits in the string S.

"""

Case study : 08

File name : cse8.py

Topics : Strings.

"""

#taking input

str=input**(“**Enter a string :”)

num=[0]\*10

for i in str:

if(ord(i)>47 and ord(i)<58):

num[ord(i)-48]+=1

for i in range(0,10):

print(i,“:”,num[i])

***Program Explanation:***

1. Take a string as input.

2. take temp array to store count of numbers.

3. For every character in string, if ascii value is between 47 and 58 increase count of acsii value –48 in array.

4. print all array elements.

5. Exit.

***Runtime Test Cases:***

1. Enter a string:77150

0:1

1:1

2:0

3:0

4:0

5:1

6:0

7:2

8:0

9:0

1. Enter a string:77150

0:1

1:1

2:0

3:0

4:0

5:1

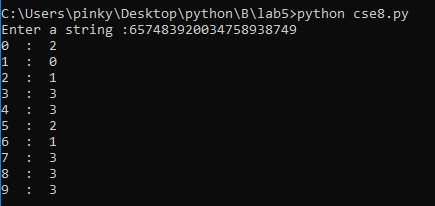
6:0

7:2

8:0

9:0

**Output:**



**Result:**

The python script to count numbers is successfully done.

***Case Study 09:***

***Problem Statement:***

The program receives 3 English words inputs from user:

These three words will be read one at a time, in three separate line

The first word should be changed like all vowels should be replaced by $

The second word should be changed like all consonants should be replaced by #

The third word should be changed like all char should be converted to upper case

Then concatenate the three words and print them

Other than these concatenated words, no other characters/string should or message should be written to Output.

***Problem Solution:***

1. Take three strings as input.

2. For 1st string if character is a or e or i or o or u replace it $

3. For 2nd string if character is not a or e or i or o or u replace it #

4. For 3rd string if character is ascii value is not between 47 and 58 make it uppercase.

5. Print all three using concatenation.

6. Exit.

***Program/Source Code:***

#Python program to change strings

"""

Case study : 09

File name : cse9.py

Topics : String

"""

#taking input

str1=input(“Enter 1st string:”)

str2=input(“Enter 2nd string:”)

str3=input(“Enter 3rd string:”)

s1=“”

s2=“”

s3=“”

for i in str1:

if(i==‘a’ or i==‘e’ or i==‘i’ or i==‘o’ or i==‘u’):

s1+=“$”

else:

s1+=i

for i in str2:

if(i!==‘a’ and i!==‘e’ and i!==‘i’ and i!==‘o’ and i!==‘u’ and i!= ‘’):

s2+=“#”

else:

s2+=i

for i in str3:

if(not(ord(i)>47 and ord(i)<58)):

s3+=i.upper()

else:

s3+=i

print(s1,“”,s2, “”,s3, “”)

***Program Explanation:***

1. Take three strings as input.

2. For 1st string if character is a or e or i or o or u add $ to s1 else add the character

3. For 1st string if character is not a or e or i or o or u add # to s2 else add the character

4.For 3rd string if character is ascii value is not between 47 and 58 add uppercased character to s3 else add the character.

5. Print all three using concatenation

6. Exit.

***Runtime Test Cases:***

1. Enter 1st string : Hello

Enter 2nd string: All Gud

Enter 3rd string: 9ice

H$ll$ ### #u# 9ICE

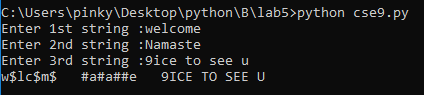
1. Enter 1st string : Hi

Enter 2nd string: Namasthe

Enter 3rd string: how 6 u

H$ #a#a###e HOW 6 U

**Output:**



**Result:**

The python script to change string is successfully done.